50103-509

CHANNEL PROCESSOR USING REDUCED COMPLEXITY LDPC DECODER

RELATED APPLICATION

[01] This application contains subject matter related to the subject matter disclosed in Provisional U.S. Patent Application Serial No. 60/415,151, filed on September 30, 2002, entitled "MAGNETIC RECORDING CHANNEL PROCESSOR USING REDUCED COMPLEXITY PR4-LDPC DECODER".

FIELD OF THE INVENTION

[02] This invention relates to methods and systems for decoding data using efficient processing architectures.

DESCRIPTION OF RELATED ART

[03] In the data recording industry, there is an ongoing effort to increase the amount of information that can be stored and retrieved in various storage media. Unfortunately, increasing the recording density on a given magnetic medium will cause a decrease in the Signal-to-Noise Ratio (SNR) of any data resident on the medium, which will subsequently result in an increased Bit-Error Rate (BER) for any detection system usable to recover such resident data.

[04] Fortunately, the performance of digital storage systems (as well as communication systems) suffering from an imperfect SNR can be significantly improved by the use of any number of error correction code schemes. As a result, most, if not all, recording (and communication) systems use some form of error correction coding, which generally involves systematically adding redundant information to a stream of data to insure that individual bit

errors generated during a particular write/read/transmission operation can be detected and corrected.

[05] In recent years, iterative correction codes have increasingly replaced the more traditionally used block and convolutional correction codes. While iterative codes, such as turbo codes and low-density parity-check (LDPC) codes, have shown very good performance for magnetic storage systems, such correction codes must of course be iteratively decoded. Unfortunately, iterative decoders often require substantial and complex computational power, and thus relatively expensive circuitry, to operate. Accordingly, improved methods and systems related to iterative decoding techniques are desirable.

SUMMARY OF THE INVENTION

[06] As described herein, an apparatus for decoding a stream of data recovered from a magnetic medium can use a specially designed low-density parity check device coupled to the magnetic medium.

[07] In various embodiments, the low-density parity check device can process data using a special low-density parity check matrix having a size of two-hundred seventy-two rows by four-thousand six-hundred and twenty-four columns. By configuring the matrix such that the matrix is formed of seventeen sub-matrices of two-hundred seventy-two rows by two-hundred seventy-two columns, only the locations of the non-zero entries in the first row of the first sub-matrix will be required to be actually generated and stored because of the structured relationship among the seventeen sub-matrices and the property that each row in a sub-matrix is the cyclic-shift of the previous row in the same sub-matrix, thus saving valuable memory resources. By then further constraining each sub-matrix to include seventeen sub-portions with each sub-portion constructed such that no two columns within a sub-portion will have a common location containing a non-zero entry, even greater processing advantages are gained.

[08] In other embodiments, the parity check device can include a checks-to-bits device that determines the minimum-entry for a particular row of the low-density parity check matrix, as well as determine the second-minimum-entry and a sign value for the same

particular row. By coupling this checks-to-bits device with a bits-to-checks device, a sub-decoder is formed that requires as few as twelve iterations to produce data having an acceptably low error rate.

[09] In still other embodiments, the data on the magnetic medium is formed using a number of data blocks with each data block including a data field, a sync field and a tone field. The decoding apparatus can accordingly incorporate a tone detector that detects a frequency in the tone fields to provide framing information for a partial response signaling device, thus simplifying processing. Further, the decoding device can incorporate a sync detector that detects a known sequence in the sync fields to provide alignment information for aligning data for the low-density parity check device.

[10] Other features and advantages of the present invention will become apparent in the following drawings and descriptions.

DESCRIPTION OF THE DRAWINGS

[11] The invention is described in detail with regard to the following figures, wherein like numerals reference like elements, and wherein:

[12] Figure 1 is a block diagram of a magnetic storage system according to the present invention;

[13] Figure 2 depicts the data format used in the storage system of Figure 1 according to the present invention;

[14] Figure 3 is a block diagram of the decoder of Figure 1 according to the present invention;

[15] Figure 4 is a block diagram of the PR4 channel decoder of Figure 3;

[16] Figure 5 is a block diagram of the LDPC decoder of Figure 3 according to the present invention;

[17] Figure 6 is a block diagram of the LDPC sub-decoder of Figure 5 according to the present invention;

[18] Figure 7 is a block diagram of the bits-to-checks portion of Figure 6;

[19] Figure 8 is a block diagram of the checks-to-bits portion of Figure 6 according to the present invention; and

[20] Figures 9A-9C depict a low-density parity check matrix according to the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[21] Figure 1 is a block diagram of a data storage system 100 according to the present invention. As shown in Figure 1, the data storage system 100 includes a data source 110, an encoder 120, a recording medium 130 having a read/write head 132, a decoder 140 and a data sink 150.

[22] In a first mode of operation, an amount of digital user data is transferred from the data source 110 to the encoder 120. The encoder 120, in turn, can condition and format the received data using a number of operations, such as a Run-Length Limiting (RLL) encoding operation, a Low-Density Parity Check (LDPC) encoding operation and a data-bit shuffling operation, to form blocks of data. Various tone and synchronization (sync) signals can then be added to the data blocks, and the resultant modified data blocks can then be processed using a pre-coder and Partial Response signaling (PR4) encoder as is well known in the digital data recording industry.

[23] As various blocks of encoded data are produced by the encoder 120, the encoder 120 can feed the data blocks to the read/write head 132. Assuming that the read/write head 132 is properly configured, energized and provided with a motion relative to the recording medium 130, the read/write head 132 will transfer the various data blocks onto the recording medium 130.

[24] In a second mode of operation, data blocks residing on the recording medium 130 can be read using the read/write head 132, and provided to the decoder 140. The decoder 140, in turn, can extract user data from the data blocks using a number of operations described below, and provide the user data to the data sink 150.

[25] The exemplary data source 110 can be any known or later developed source that is capable of providing digital data to the encoder 120. Similarly, the exemplary data sink 150

can be any known or later developed device that is capable of receiving data from the decoder 140.

[26] The exemplary encoder 120 and decoder 140 are implemented using dedicated VLSI logic. However, the particular form of the encoder 120 or decoder 140 can vary according to the particular design needs of a given recording system to include any combination of electronic and/or optical hardware processing without departing from the spirit and scope of the present invention.

[27] The exemplary recording medium 130 is a magnetic-base substrate and the exemplary read/write head 132 is a transducer capable of receiving electrical signals and converting the electronic signals to magnetic fields capable of imprinting digital data on the magnetic recording medium 130, and similarly capable of sensing magnetic fields residing on the magnetic recording medium 130 and converting the sensed magnetic fields to electrical signals. However, it should be appreciated that the particular form of the recording medium 130 and read/write head 132 can vary to embody any number of magnetic recording systems, such as magnetic-based hard disks, floppy disk, tape-based systems and the like, without departing from the spirit and scope of the present invention.

[28] Further, the recording medium 130 and read/write head 132 can in various embodiments take various forms to accommodate any number of optical recording systems, or even be replaced with a data transmission medium for use in any number of communication systems, such as wireless and optical communication systems, without departing from the spirit and scope of the present invention.

[29] Figure 2 depicts the data format used in the exemplary system 100 of Figure 1. As discussed above, user data provided by the data source 110 is encoded, then formed into blocks having various tone and synchronization (sync) signals added. As shown in Figure 2, the exemplary data format includes a stream of data 200 partitioned into successive data blocks 210, with each data block 210 including a tone field 212 containing a signal of a given frequency, a sync field 214 containing an identifiable sequence of digital bits and a data field 216 containing encoded user data.

[30] Figure 3 is a block diagram of the decoder 140 of Figure 1. As shown in Figure 3, the decoder 140 includes a pre-processor 310, a tone-detector 312, a Finite-Impulse-Response (FIR) filter 314, an Automatic Gain Control (AGC) device 316, an Interpolate Timing Recovery (ITR) device 318, a PR4 channel decoder 320, a sync detector 322, an un-shuffler 324, a LDPC decoder 326 and an RLL decoder 328. While the exemplary decoder 140 is implemented using various VLSI logic circuits, it should be appreciated that the various devices 302-328 can be implemented using any combination of signal processing hardware, sequential instruction devices and dedicated logic, without departing from the spirit and scope of the present invention.

[31] In operation, a digitized signal can be received by the pre-processor 310 from an external source, such as a read/write head of a storage media or a receiver of a communication system. The pre-processor 310 can then correct for any asymmetry of the signal caused by any number of external factors, e.g., the non-linearity of a magneto-resistive read/write head. Once the received signal is corrected, the corrected signal can be provided to both the tone detector 312 and FIR 314.

[32] The FIR 314 can receive the corrected signal, perform an equalization process tailored to the requirements of the PR4 channel decoder 320 and provide the equalized signal to the ITR 318. The exemplary FIR 314 uses twenty-four taps with respective programmable weighting coefficients to provide a sum-of-weighted-products output. However, the particular size and makeup of the FIR 314 can vary as required without departing from the spirit and scope of the present invention.

[33] As with the FIR 314, the tone detector 312 can similarly receive the corrected signal from the pre-processor 310. However, the particular function of the tone detector 312 is to detect a signal having a specified frequency, such as the signal of the tone field 212 of Figure 2. Once the specified frequency is detected, the tone detector 312 can act as a switch for the AGC 316 and ITR 318 to enter a fast acquisition mode, as well as provide framing information to the PR4 channel decoder 320. Still further, the tone detector 312 can be used to qualify the validity of a sync signal given that (according to Figure 2) sync signals are to

immediately follow tone signals. That is, by requiring a valid tone to immediately precede a recognized sync pattern, false sync detection can be avoided.

[34] The exemplary tone detector 312 uses a peak detection scheme that calculates the moving average of the time periods of four peaks, and compares the moving average to an acceptable range of qualifying time periods. However, it should be recognized that the particular form of the tone decoder 312 can vary to implement any number of well known or later developed tone/frequency detection techniques without departing from the spirit and scope of the present invention.

[35] The ITR 318 can receive the equalized signal from the FIR 314 and, once activated by an activation signal from the tone detector 312, the ITR 318 can recover timing and frequency information from the equalized signal, and provide the recovered timing and frequency information to the PR4 channel decoder 320.

[36] The AGC 316 can similarly receive the equalized signal (passed on via the ITR 318) and, once activated by an activation signal from the tone detector 312, the AGC 316 can correct for any gain change of the received signal. Once the AGC 316 appropriately performs its gain-compensation operation, the resultant gain-compensated signal can be provided to the PR4 channel decoder 320 via the ITR 318.

[37] The PR4 channel decoder 320 can receive the gain-compensated equalized signal from the ITR 318 and, using framing information provided by the tone decoder 312, the PR4 channel decoder 320 can perform a partial response decoding operation. The exemplary PR4 channel decoder 320 uses a Bahl, Cocke, Jelinek and Raviv (BCJR) algorithm, which is a known variant of the maximum-a-posteriori (MAP) decoding approach. The BCJR algorithm (also known as the forward-backward algorithm) can provide a trellis-based decoder solution with the exemplary overall trellis operating according to the code $(1\text{-}D^2)/(1 \oplus D^2)$. However, it should be appreciated that in other embodiments, the PR4 channel decoder 320 can use other available trellis codes, and in still other embodiments the PR4 channel decoder 320 can use any number of useful non-trellis techniques.

[38] The exemplary PR4 channel decoder 320 operates under the assumption that blocks of user data will be separated by a particular tone. Accordingly, the PR4 channel decoder 320

can use the tone fields 212 of Figure 2 as framing information, i.e., start and stop signals, thus providing accurate timing information for the PR4 channel decoder 320. Once the PR4 channel decoder 320 has performed its decoding operation, the PR4 decoded data can be provided to the un-shuffler 324 and the sync detector 322.

[39] The sync detector 322 can receive the PR4 decoded data along with the pre-processed data provided by the pre-processor 310 (passed on via the tone detector 312), and detect a sync code, i.e., a predetermined bit sequence/pattern of digital bits in the sync field 214 of Figure 2. Assuming that a bit sequence recognized as a sync code is immediately preceded by a tone signal (as qualified by the tone detector 312), the sync detector 322 can generate an alignment signal capable of aligning data for the un-shuffler 324 and LPDC decoder 326.

[40] The un-shuffler 324 can receive the PR4 decoded data and, using the alignment signal from the sync detector 322, extract parity bits embedded in a data field and put the parity bits in an appropriate order for the LDPC decoder 326.

[41] The LDPC decoder 326 can receive the un-shuffled data, as well as the alignment signal from the sync detector 322 (via the un-shuffler 324), and perform an LDPC decoding operation. Once an LDPC operation is performed for a block of data, the LDPC decoded data is provided to the RLL decoder 328.

[42] The exemplary LDPC decoder 326 uses a LDPC code having a rate of 16/17, and also uses a special LDPC matrix having a size of 272 rows by 4624 columns (depicted in Figure 9A as matrix 910) with the LDPC matrix 910 having a constant column weight of 3 and a constant row weight of 51. The particular 272-by-4624 dimensions of the low-density parity check matrix 910 take on special significance and provide substantial advantage for a 16/17 LDPC code rate in view of the chosen column and row weights as the 272-by-4624 dimension can allow for greatly reducing various processing and memory requirements.

[43] For example, the LDPC matrix 910 of Figure 9A can be configured to be cyclic to repeat itself every so many columns such that the LDPC matrix 910 can be divided into seventeen square sub-matrices 910-1 ... 910-17 (as shown in Figure 9B) with each of the sub-matrices 910-1 ...910-17 having a size of 272 rows by 272 columns. As each sub-matrix 910-1 ... 910-17 can be computed from one another, it becomes evident that an entire 272-by-

4624 matrix need not be stored as all the information in the 272-by-4624 matrix can be contained in the first row of the first 272-by-272 sub-matrix. In view of such advantages, it should be appreciated that the dimensions of an LDPC matrix can in other embodiments strategically vary with varying LDPC code rates and matrix weights with similar dynamics providing similar advantages.

[44] Returning to Figure 3, the RLL decoder 328 can receive the LDPC decoded data, and perform an RLL decoding operation to extract original user data. The exemplary RLL decoder 328 uses an RLL code that decodes data byte-by-byte (or word-by-word) independently to minimize error propagation. However, the particular form of the RLL code can vary as required without departing from the spirit and scope of the present invention.

[45] Figure 4 is a block diagram of the PR4 channel decoder 320 of Figure 3. As shown in Figure 4, the PR4 channel decoder 320 includes an interleaver 410, a number of PR4-channel sub-decoders 420, 422, 430 and 432, and a de-interleaver 440.

[46] In operation, the interleaver 410 can receive data from a device, such as the ITR device 318 of Figure 3, separate the received data into even and odd bits, provide the odd bits to sub-decoders 420 and 422 and provide the even bits to sub-decoders 430 and 432.

[47] As the data bits are provided to the sub-decoders 420, 422, 430 and 432, sub-decoders 420 and 430 will work as a cooperating group as will sub-decoders 422 and 432. The two groups of sub-decoders (420, 430) and (422, 432) will then work together in an interleaved fashion with each group of sub-decoders (420, 430) or (422, 432) performing the classic forward and backward calculations characteristic of BCJR algorithms (or their max-log approximations) with an optional normalization step for further ease of processing.

[48] As the sub-decoders 420, 422, 430 and 432 provide their respective decoded data to the de-interleaver 440, the de-interleaver 440 will appropriately assemble the interleaved PR4 decoded data, then output the assembled PR4 decoded data X[n].

[49] Figure 5 is a block diagram of the exemplary LDPC decoder 326 of Figure 3. As shown on Figure 5, the exemplary LDPC decoder 326 includes n-number of LDPC sub-decoders 510-n, an adding device 520 and a decision device 530. In operation, the various LDPC sub-decoders 510-1 perform an iterative operation on received data X[n] as will be

further explained below. After the last processing iteration, LDPC sub-decoder 510-n will supply its outputs R[0], R[l] and R[2] to the adding device 520. The resultant sum of the adding device 520, in turn, can be provided to the decision device 530, which can then determine and report the sign of the resultant sum.

[50] In determining the effectiveness of the LDPC decoder 326, it should be appreciated that the inventors have discovered that good performance can be achieved with as little as twelve sub-processor iterations with the understanding that further iterations can further improve data accuracy.

[51] Figure 6 is a block diagram of an LDPC sub-decoder 510. As shown on Figure 6, an LDPC sub-processor 510 can be divided into two parts, a bits-to-checks portion 610 and a checks-to-bits portion 630. Given this, it should be apparent that the decoding algorithm used by an LDPC decoder is an iterative process employing two basic kinds of computations -- bits-to-checks computations and checks-to-bits computations. As observed in Figure 6, there is a 48-sample conduit between the bits-to-checks portion 610 and the checks-to-bits portion 630, which demonstrates a high degree of parallelism available by the LDPC sub-decoder 510 of the present invention. Such parallelism can be attributed in great part to the form of the LDPC matrix.

[52] As discussed above, and demonstrated in Figures 9A and 9B, the LDPC decoder 326 can use a 272-by-4624 LDPC matrix that can be expressed in full by a much smaller 272-by-272 LDPC sub-matrix. However, by further constraining the LDPC sub-matrix (or similarly the LDPC matrix) to be divided into seventeen sub-portions 920 (as shown in Figure 9C) with each sub-portion 920 having sixteen consecutive columns each (i.e., 1-16, 17-31, 33-48. . .), and require that no two of the sixteen columns within each sub-portion 920 have any common location containing a non-zero entry, then the LDPC sub-decoder 510 can perform forty-eight simultaneous operations.

[53] Figure 7 depicts the bits-to-checks portion 610 of sub-decoder 510. As shown on Figure 6, the bits-to-checks portion 610 includes a bits-to-checks device 710 having three adding devices 712, 714 and 716 and a column-to-row routing network 750. In an iterative operation, the bits-to-checks device 710 can receive three signals R[0], R[l] and R[2] as well

as a PR4 channel decoder signal X[n], and operate on the received signals to produce three outputs Q[0]= R[l] + R[2] - X[n], Q[l]= R[0] + R[2] - X[n], and Q[2]= R[0] + R[l] - X[n]. The three outputs Q[0], Q[l] and Q[2] are then provided to the column-to-row routing network 750, which reallocates each data sample to a particular row according to the above-mentioned constraints of the LDPC matrix depicted in Figures 9A-9c. The reallocated data is then output to an external device, such as the checks-to-bits portion 630 of Figure 6.

[54] Figure 8 is a block diagram of the checks-to-bits portion 630 of Figure 6. As shown in Figure 6, the checks-to-bits portion 630 includes a checks-to-bits device 810 and a row-to-column routing network 880. The checks-to-bits device 810 itself includes an absolute value device 812, an array of comparators 814, a minimum-entry memory 816, a second-minimum-entry memory 818, a sign determining device 820, an array of exclusive-OR devices 822 and a sign-memory 824.

[55] In operation, a number of parallel samples (48 in this particular case) are received by the checks-to-bits device 810 and provided to the absolute value device 812 and the sign determining device 820.

[56] The absolute value device 812, in turn, can take the magnitude of each received sample, and provide a set of sample magnitudes to the array of comparators 814. Similarly, the sign determining device 820 can strip each received sample of its magnitude to provide a set of sample signs ("+" or "-") to the array of exclusive-OR devices 822.

[57] Upon reception of the sample magnitudes, the array of comparators 814 can then operate to find the sample for each particular row of the LDPC matrix having the smallest magnitude, i.e., the minimum-entry sample (a.k.a., the "minl" entry), and deposit the minimum-entry sample into the minimum-entry memory 816 (a.k.a., the "minl" memory). Similarly, the array of comparators 814 can operates to find the sample for each particular row of the LDPC matrix having the second to smallest magnitude, i.e., the second-minimum-entry sample (a.k.a., the "min2" entry), and deposit the second-minimum-entry sample into the second-minimum-entry memory 818 (a.k.a., the "min2-memory").

[58] Upon reception of the sample signs, the array of exclusive-OR devices 822 can operate to find the exclusive-OR product for the samples of each particular row of the LDPC

matrix, which will take the value of a positive(+) or negative(-) sign. Once determined, each sign-value can be deposited into the sign-memory 824.

[59] For the present embodiment using the 272-by-4624 LDPC matrix, each of the minl-memory 816, min2-memory 818 and sign-memory 824 can hold 272 samples. Further, for each checks-to-bits computation, there will be 51 samples in each row, and each entry of the 272 rows will only take on one of four values; ±minl or ±min2.

[60] Returning to Figure 8, once the minl-memory 816, min2-memory 818 and sign-memory 824 are appropriately updated, the memories 816, 818 and 824 can provide their content to the row-to-column routing network 880, which will reorganize the LDPC matrix in a reverse manner to the column-to-row routing network 750 of Figure 7 to produce three outputs R[0], R[I] and R[2].

[61] As discussed above, the operation of the LDPC sub-decoder 510 can be repeated as required as described by Figures 5-8 and respective text until adequate performance is acquired. As shown in Figures 1-8, the systems and methods of this invention are preferably implemented using dedicated logic or other integrated circuits. However, the systems and methods can also be implemented using any combination of one or more general purpose computers, special purpose computers, program microprocessors or microcontroller and peripheral integrating circuit elements, hardware electronic or logic circuits such as application specific integrated circuits (ASICs), discrete element circuits, programmable logic devices such as PLAs, FPGAs, PALs or the like. In general, any device on which exists a finite state machine capable of implementing the various elements of Figures 1-8 and the matrix of Figures 9A-9C can be used to implement the training sequence functions.

[62] The foregoing describes various embodiments that provide a number of processing advantages in that both the total number of mathematical operations and required memory are reduced due at least in part to the particular dimensions and restraints of the low-density parity check matrix. Further processing advantages are gained by virtue of the architecture of the sub-decoder, including the minimum-entry approach of the sub-decoder 's checks-to-bits device. In addition to reducing memory and processing, the inventive use of sync fields and

tone fields provides framing information for more reliable data alignment for the low-density parity check device, thus again simplify processing.

[63] The foregoing description of various embodiments have been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed, and modifications and variations are possible in light of the above teachings or may be acquired from practice of the invention. The embodiments were chosen or described in order to explain the principles of the invention and enable one of ordinary skill in the art to utilize this systems with various modifications as would be suited to a particular use as contemplated. It is intended that the scope of the various embodiments be defined by the claims appended hereto, and their equivalents.